

Nov 30, 17 23:54	MasterMind.as	Page 5/18
<pre> ## ;##### ##### ##### ##### ;##### ; ;##### # ESCREVE_JOGADA_LCD: MOV M[MELHOR_JOGADA_LCD], R4 MOV R6, POSICAO_LCD3 MOV M[LCD_CURSOR], R6 INC R4 ADD R4, 48 MOV M[LCD_ESCREVE], R4 MOV R1, R4 RET ESCREVE_LCD: PUSH R2 PUSH R3 PUSH R6 MOV R6, POSICAO_LCD1 MOV M[LCD_CURSOR], R6 MOV R2, Frase_LCD1 CALL CICLO_ESCRITA_LCD MOV R6, POSICAO_LCD2 MOV M[LCD_CURSOR], R6 MOV R2, Frase_LCD2 CALL CICLO_ESCRITA_LCD POP R6 POP R3 POP R2 RET CICLO_ESCRITA_LCD: PUSH R6 ESCRITA_LCD_LOOP: MOV R3, M[R2] CMP R3, EOT BR.Z RETURN_ESCRITA_LCD MOV M[LCD_ESCREVE], R3 INC R2 INC R6 MOV M[LCD_CURSOR], R6 BR RETURN_ESCRITA_LCD RETURN_ESCRITA_LCD: BR POP R6 RET ESCREVE_SEQ_JOG: PUSH R2 AND R2, F000h CMP R2, R0 POP R2 BR.NZ SEGUNDO_DIGITO ROR R6, 4 ADD R2, R6 MOV R6, 0 RET SEGUNDO_DIGITO: PUSH R2 </pre>		

Nov 30, 17 23:54	MasterMind.as	Page 6/18
<pre> AND R2, 0F00h CMP R2, R0 POP R2 BR.NZ TERCEIRO_DIGITO ROR R6, 8 ADD R2, R6 MOV R6, 0 RET TERCEIRO_DIGITO: PUSH R2 AND R2, 00F0h CMP R2, R0 POP R2 BR.NZ QUARTO_DIGITO ROL R6, 4 ADD R2, R6 MOV R6, 0 RET QUARTO_DIGITO: ADD R2, R6 MOV R6, R2 RET ;##### ; ;##### # APAGA_DISPLAYS: MOV M[DSP1], R0 MOV M[DSP2], R0 MOV M[DSP3], R0 MOV M[DSP4], R0 RET CONTAGEM_JOGADAS: INC R4 PUSH R2 PUSH R4 MOV R2, 10 DIV R4, R2 MOV M[DSP4], R2 MOV M[DSP3], R4 POP R4 POP R2 CMP R4, 12 JMP.Z DERROTA ;Apos 12 tentativas(j ogadas) o jogo termina RET ;##### ; ;##### # CHAVE_ALEATORIA: MOV R3, 4 MOV R5, 0 CICLO: PUSH R1 GERADOR_ALEATORIO: AND R1, 0001h CMP R0, R1 POP R1 BR.NZ ALTERNATIVA ROR R1, 4 BR RESTO ALTERNATIVA: XOR R1, Mascara ROR R1, 4 </pre>		

Nov 30, 17 23:54	MasterMind.as	Page 7/18
RESTO:	MOV R2, 5 PUSH R1 AND R1, FFF0h MOV R4, R1 POP R1 AND R1, 000Fh DIV R1, R2 INC R2 ADD R4, R2 MOV R1, R4 ROR R1, 4 INC R5 CMP R3, R5 BR.NZ RESTO RET	
;#####		
; TEMPORIZADOR/INSERIR SEQUENCIA JOGADOR #		
;#####		
INTRODUZIR_TENT:	MOV R6, 5 PUSH R5 MOV R5, 5 MOV M[TEMP], R5 MOV R5, 1b MOV M[CTRL_TEMP], R5 ENI MOV R5, FFFFh MOV M[LED], R5 MOV M[LEDS], R5 POP R5 PUSH R4 MOV R4, M[LED] CMP R4, 0 POP R4 BR.Z FIM_DO_TEMPO	
INTRODUZIR_LOOP:		
FIM_DO_TEMPO:	CMP R6, R2 BR.NZ INTRODUZIR_LOOP RET	
;#####		
;CONTAGEM X: Rotina que compara os d��gitos que est��o #		
; na mesma posicao entre a sequencia #		
; do jogador e a gerada aleatoriamente #		
;#####		
COMPARACAO_X:	PUSH R1 PUSH R2 AND R1, 000Fh AND R2, 000Fh CMP R1, R2 POP R2 POP R1 CALL.Z MOV_X RET	
CONTAGEM_X:	CALL COMPARACAO_X ROR R1, 4 ROR R2, 4 CALL COMPARACAO_X	

Nov 30, 17 23:54	MasterMind.as	Page 8/18
	ROR R1, 4 ROR R2, 4 CALL COMPARACAO_X ROR R1, 4 ROR R2, 4 CALL COMPARACAO_X ROR R1, 4 ROR R2, 4 RET	
;*****		
; * IMPLEMENTACAO DOS X's: *		
; * Sub-rotina que coloca em R3 os X's encontrados *		
;*****		
MOV_X:	PUSH R3 AND R3, F000h CMP R3, R0 POP R3 BR.Z MOV_X4 PUSH R3 AND R3, 0F00h CMP R3, R0 POP R3 BR.Z MOV_X3 PUSH R3 AND R3, 00F0h CMP R3, R0 POP R3 BR.Z MOV_X2 PUSH R3 AND R3, 000Fh CMP R3, R0 POP R3 BR.Z MOV_X1	
MOV_X4:	ADD R3, 3000h CALL RM_X RET	
MOV_X3:	ADD R3, 0300h CALL RM_X RET	
MOV_X2:	ADD R3, 0030h CALL RM_X RET	
MOV_X1:	ADD R3, 0003h CALL RM_X RET	
;*****		
; * REMOCAO DOS PARES: *		
; * Sub-rotina que remove os digitos iguais que estejam *		
; * nas mesmas posicoes das duas sequencias *		
;*****		
RM_X:	AND R1, FFF0h	

Nov 30, 17 23:54	MasterMind.as	Page 11/18
<pre> ;# CONTAGEM NULO: Rotina que verifica quais sao os # ;# digitos que sobraram na sequencia # ;# gerada aleatoriamente # ;##### </pre>		
VERIFICA_NULO:	PUSH R1	
	AND R1, 000Fh	
	CMP R1, R0	
	POP R1	
	CALL.NZ MOV_NULO	
	RET	
CONTAGEM_NULO:	CALL VERIFICA_NULO	
	ROR R1, 4	
	CALL VERIFICA_NULO	
	ROR R1, 4	
	CALL VERIFICA_NULO	
	ROR R1, 4	
	RET	
<pre> ;***** ;* IMPLEMENTACAO DOS NULOS * ;* Sub-Rotina que coloca em R3 o numero de digitos * ;* que nao eram iguais nas duas sequencias * ;***** </pre>		
MOV_NULO:	PUSH R3	
	AND R3, F000h	
	CMP R3, R0	
	POP R3	
	BR.Z MOV_NULO1	
	PUSH R3	
	AND R3, 0F00h	
	CMP R3, R0	
	POP R3	
	BR.Z MOV_NULO2	
	PUSH R3	
	AND R3, 00F0h	
	CMP R3, R0	
	POP R3	
	BR.Z MOV_NULO3	
	PUSH R3	
	AND R3, 000Fh	
	CMP R3, R0	
	POP R3	
	BR.Z MOV_NULO4	
MOV_NULO1:	ADD R3, 1000h	
	RET	
MOV_NULO2:	ADD R3, 0100h	
	RET	
MOV_NULO3:	ADD R3, 0010h	
	RET	
MOV_NULO4:	ADD R3, 0001h	
	RET	

Nov 30, 17 23:54	MasterMind.as	Page 12/18
<pre> ;##### ;## ## ;##### </pre>		
CICLO_ESCRITA:	PUSH R6	
CICLO_ESCRITA_LOOP:	MOV R3, M[R2]	
	CMP R3, EOT	
	BR.Z RETURN_ESCRITA	
	MOV M[IO_ESCREVE], R3	
	INC R2	
	INC R6	
	MOV M[IO_CURSOR], R6	
RETURN_ESCRITA:	BR CICLO_ESCRITA_LOOP	
	POP R6	
	RET	
<pre> ;***** ;* MAIN MENU: Rotina que escreve na janela de texto * ;* a pagina inicial do Programa * ;***** </pre>		
MAIN_MENU:	PUSH R6	
	PUSH R2	
MASTERMIND:	MOV R6, POSICAO_TITULO	
	escreve a string do Titulo do jogo	
	MOV M[IO_CURSOR], R6	
	MOV R2, Mastermind1	
	CALL CICLO_ESCRITA	
	ADD R6, 0100h	
	MOV M[IO_CURSOR], R6	
	MOV R2, Mastermind2	
	CALL CICLO_ESCRITA	
	ADD R6, 0100h	
	MOV M[IO_CURSOR], R6	
	MOV R2, Mastermind3	
	CALL CICLO_ESCRITA	
	ADD R6, 0100h	
	MOV M[IO_CURSOR], R6	
	MOV R2, Mastermind4	
	CALL CICLO_ESCRITA	
AUTORES:	MOV R6, POSICAO_COPYRIGHT	
	escreve a string 'Copyright'	
	MOV M[IO_CURSOR], R6	
	MOV R2, COPYRIGHT	
	CALL CICLO_ESCRITA	
PRIMA_COMECAR:	MOV R6, POSICAO_PRIMA_PARA_COMECAR	
	escreve a string de comecar	
	MOV M[IO_CURSOR], R6	
	MOV R2, COMECAR	
	CALL CICLO_ESCRITA	
	POP R2	
	POP R6	
	RET	
<pre> ;***** ;* ESTRUTURA TABELA: Rotina que escreve na janela de texto * ;* a tabela e o numero total de tentativas * ;* que o jogador tem * ;***** </pre>		

Nov 30, 17 23:54	MasterMind.as	Page 13/18
NOVA_LINHA_TABELA:	ADD	R6, 0200h MOV M[IO_CURSOR], R6 RET
TABELA:	PUSH	R2 PUSH R3 PUSH R4 PUSH R5 PUSH R6 MOV R4, 23 MOV R6, POSICAO_TABELA
N_COLUNAS:	MOV	M[IO_CURSOR], R6 MOV R2, ESTRUTURA_TABELA CALL CICLO_ESCRITA INC R5 ADD R6, 0100h CMP R5, R4 BR.NZ N_COLUNAS MOV R6, POSICAO_NUM_TENTATIVA MOV M[IO_CURSOR], R6 MOV R2, TENTATIVA_1 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_2 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_3 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_4 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_5 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_6 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_7 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_8 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_9 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_10 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA

Nov 30, 17 23:54	MasterMind.as	Page 14/18
	MOV	R2, TENTATIVA_11 CALL CICLO_ESCRITA CALL NOVA_LINHA_TABELA MOV R2, TENTATIVA_12 CALL CICLO_ESCRITA POP R6 POP R5 POP R4 POP R3 POP R2 RET
; ***** ; * REPRESENTACAO SEQUENCIA INSERIDA PELO JOGADOR E O RESPETIVO RESULTADO * ; *****		
REPRES_RESULTADO:	CMP	R4, 0 CALL.Z POSICAO_RESULTADOS CMP R4, 0 BR.Z RESULTADO__1 ADD R5, 0200h
RESULTADO__1:	PUSH	R5 ROL R3, 4 CALL ESCREVE_RESULTADO ROL R3, 4 CALL ESCREVE_RESULTADO ROL R3, 4 CALL ESCREVE_RESULTADO ROL R3, 4 CALL ESCREVE_RESULTADO POP R5 RET
POSICAO_RESULTADOS:	MOV	R5, POSICAO_RESULTADO MOV M[IO_CURSOR], R5 RET
ESCREVE_RESULTADO:	PUSH	R3 AND R3, 000Fh MOV M[IO_CURSOR], R5 CMP R3, 0003h CALL.Z ESCREVE_X CMP R3, 0002h CALL.Z ESCREVE_O CMP R3, 0001h CALL.Z ESCREVE_NULO POP R3 ADD R5, 2 MOV M[IO_CURSOR], R5 RET
ESCREVE_X:	MOV	R6, 'X' MOV M[IO_ESCREVE], R6 RET
ESCREVE_O:	MOV	R6, 'O' MOV M[IO_ESCREVE], R6 RET

Nov 30, 17 23:54	MasterMind.as	Page 15/18
ESCREVE_NULO:	MOV R6, '-'	
	MOV M[IO_ESCREVE], R6	
	RET	
;=====		
REPRES_CODIGO_JOG:	CMP R4, 0	
	CALL.Z POSICAO_CODIGOS	
	CMP R4, 0	
	BR.Z CODIGO__1	
	ADD R7, 0200h	
CODIGO__1:	PUSH R7	
	ROL R2, 4	
	CALL ESCREVE_CODIGO	
	ROL R2, 4	
	CALL ESCREVE_CODIGO	
	ROL R2, 4	
	CALL ESCREVE_CODIGO	
	ROL R2, 4	
	CALL ESCREVE_CODIGO	
	POP R7	
	RET	
POSICAO_CODIGOS:	MOV R7, POSICAO_CODIGO_JOGADOR	
	MOV M[IO_CURSOR], R7	
	RET	
ESCREVE_CODIGO:	PUSH R2	
	AND R2, 000Fh	
	ADD R2, 48	
	MOV M[IO_CURSOR], R7	
	MOV M[IO_ESCREVE], R2	
	POP R2	
	ADD R7, 2	
	MOV M[IO_CURSOR], R7	
	RET	
;*****		
;* DERROTA: Rotina que escreve na janela *		
;* de texto que o jogador perdeu *		
;*****		
ESCREVE_DERROTA:	PUSH R2	
	PUSH R6	
	MOV R6, POSICAO_DERROTA	
	MOV M[IO_CURSOR], R6	
	MOV R2, FIM_DO_JOGO_MAU	
	CALL CICLO_ESCRITA	
	CALL ESCREVE_RECOMECAR	
	POP R6	
	POP R2	
	RET	
;*****		
;* VITORIA: Rotina que escreve na janela *		
;* de texto que o jogador ganhou *		
;*****		
ESCREVE_VITORIA:	PUSH R2	
	PUSH R6	
	MOV R6, POSICAO_VITORIA	
	MOV M[IO_CURSOR], R6	
	MOV R2, FIM_DO_JOGO_BOM	
	CALL CICLO_ESCRITA	

Nov 30, 17 23:54	MasterMind.as	Page 16/18
	CALL ESCREVE_RECOMECAR	
	POP R6	
	POP R2	
	RET	
;*****		
;* RECOMECAR JOGO: Rotina que escreve na janela *		
;* de texto indicacoes para *		
;* recomecar o jogo *		
;*****		
ESCREVE_RECOMECAR:	PUSH R2	
	PUSH R6	
	MOV R6, POSICAO_PRIMA_PARA_RECOMECAR1	
	MOV M[IO_CURSOR], R6	
	MOV R2, TRY_AGAIN	
	CALL CICLO_ESCRITA	
	MOV R6, POSICAO_PRIMA_PARA_RECOMECAR2	
	MOV M[IO_CURSOR], R6	
	MOV R2, TRY_AGAIN2	
	CALL CICLO_ESCRITA	
	PUSH R5	
	MOV R5, 0000h	
	MOV M[LED], R5	
	MOV M[LEDS], R5	
	POP R5	
	POP R6	
	POP R2	
	RET	
;*****		
;* APAGAR JANELA: Rotina que apaga a janela *		
;* de texto toda *		
;*****		
APAGAR_JANELA:	PUSH R2	
	PUSH R4	
	PUSH R6	
	MOV R4, 1850h	
	MOV R6, 0000h	
	MOV R2, ' '	
CICLO_APAGAR:	INC R6	
	MOV M[IO_CURSOR], R6	
	MOV M[IO_ESCREVE], R2	
	CMP R6, R4	
	BR.NZ CICLO_APAGAR	
	POP R6	
	POP R4	
	POP R2	
	RET	
;#####		
;#####		
;#####		
;##		
;##		
;##		

Nov 30, 17 23:54	MasterMind.as	Page 17/18
INICIO DO JOGO		
##		
;##		
##		
;#####		
#####		
#####		
INICIO_JOGO:	CALL	MAIN_MENU
INICIO_NOVO_JOGO:	CALL	APAGA_DISPLAYS
	MOV	R1, 0 ;
	MOV	R2, 0 ;
	MOV	R3, 0 ;Inicializacao da joga
da com todos os registos a 0		
	MOV	R4, 0 ;
	MOV	R5, 0 ;
	MOV	R6, 0 ;
	ENI	
	PUSH	R6
PRIMA_IA:	INC	R1 ;Enquanto nao for premido o in
erruptor A o programa incrementa R1 (ponto de partida para gerar a chave aleato		
ria)		
	CMP	R6, B00Bh ;Rotina que na
o permite o programa avançar enquanto o interruptor A nao for pressionado		
	BR.NZ	PRIMA_IA
	POP	R6
	DSI	
	PUSH	R2
	PUSH	R3
	PUSH	R4
	PUSH	R5
	CALL	CHAVE_ALEATORIA ;Gerador da sequ
encia aleatoria e colocacao da mesma em R1		
	POP	R5
	POP	R4
	POP	R3
	POP	R2
	CALL	APAGAR_JANELA
	CALL	TABELA
	PUSH	R5
	MOV	R5, FFFFh
	MOV	M[LEDS], R5
	POP	R5
INTRODUZIR_R2:	ENI	
	PUSH	R6
	CALL	INTRODUZIR_TENT ;Rotina que nao
permite o programa avançar enquanto o jogador nao introduzir a sua sequencia		
	POP	R6
	PUSH	R4
	MOV	R4, M[LED]
	CMP	R4, R0
	POP	R4
	JMP.Z	DERROTA
	DSI	
	CMP	R2,R1
	JMP.Z	VITORIA ;Caso a sequecia gerada
aleatoriamente e a introduzida pelo jogador sejam iguais, o jogo segue um ramo '		

Nov 30, 17 23:54	MasterMind.as	Page 18/18
a parte		
	MOV	R3, R0
	PUSH	R1
	PUSH	R2
	CALL	ESCREVE_LCD
	CALL	CONTAGEM_X
	CALL	CONTAGEM_O
	CALL	CONTAGEM_NULO
	POP	R2
	POP	R1
	CALL	REPRES_RESULTADO ;Representaca
o do resultado da comparacao entre a sequencia secreta e a do jogador na janela		
de texto		
	CALL	REPRES_CODIGO_JOG ;Representaca
o da sequencia inserida pelo jogador na janela de texto		
	CALL	CONTAGEM_JOGADAS ;Contagem do
numero de jogadas que o jogador efetua num jogo		
	MOV	R2, R0
	JMP	INTRODUZIR_R2 ;Apos cada jogada
, o jogo reinicia, partindo do ponto em que e' pedido a insercao de uma sequenci		
a		
;#####		
#####		
#####		
;##		
	##	
;##		
	FIM DO JOGO	
	##	
;##		
	##	
;#####		
#####		
#####		
DERROTA:	CALL	ESCREVE_DERROTA
	JMP	INICIO_NOVO_JOGO
VITORIA:	MOV	R3, 3333h
	CALL	REPRES_RESULTADO
	CALL	REPRES_CODIGO_JOG
	CALL	ESCREVE_VITORIA
	PUSH	R5
	MOV	R5, M[MELHOR_JOGADA_LCD]
	CMP	R4, R5
	POP	R5
	CALL.N	ESCREVE_JOGADA_LCD
	JMP	INICIO_NOVO_JOGO